International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

SMART WIRELESS WATER METER WITH WEB DB IOT

Rohith M[1] , Sambasivareddy P[1] , Lakshmi Ajay P[1] , Dr. Padmaloshani P[2]

[1]UG Scholar, Department of Electronics and Communication Engineering, Muthayammal Engineering College – Rasipuram, Namakkal(D.t), Tamilnadu - INDIA

[2]Assosicate Prof., Department of Electronics and Communication Engineering, Muthayammal Engineering College – Rasipuram, Namakkal(D.t), Tamilnadu - INDIA

ABSTRACT

Water is a precious need of our lives. Due to the rapid population and urbanization, water usage monitoring is a significant problem facing our society. One solution is to control, analyze, and reduce the water consumption of the houses. The emerging of the Internet of Things (IoT) concept lately in our lives has offered the opportunity to establish water usage-efficient smart devices, systems and applications for buildings and cities. Many studies have suggested designing an IoT-based smart meter system; however, the IoT sensor node has limited studies, especially in battery life. The current system provides a unique twist to the smart water meters using IoT technology, where the smart meter data is stored in the cloud. Nevertheless, the IoT sensor node has limited resources, especially in battery life. Therefore, this project aims to reduce the energy consumption during the data sending process to the cloud, extending the sensor's node battery lifetime by implementing and analyzing an efficient data collection algorithm for IoT-based smart water metering application. For a more high-efficient operation of water-usage, we have designed a smart water meter using IoT technology. As IoT-based devices have limited power resources and rely on batteries, therefore we proposed an efficient data collection algorithm to reduce the data transmission and thus we can extend the battery life. Therefore, this study aims to implement and analyze an efficient data collection algorithm for IoT-based smart metering applications in consideration with energy consumption. The system items used are PIC microcontroller, Wi-Fi-ESP8266, and water flow sensors. The applied algorithm is an efficient data collection algorithm for water meter (EDCDWM) to reduce the number of packet transmissions. Arduino performed this system's implementation, while the simulation and analysis performed by hardware model.

***Keywords** - Internet of things; smart water metering and energy consumption etc.,*

1. INTRODUCTION

In many countries, water conservation is becoming increasingly necessary as countries face a widening gap between the ever-decreasing water availability due to climate change and the rising demand for population growth. Water efficiency implies less water consumption and searching for an alternative of conventional water meters to measure the quantity and quality of water. Water utilities build daily demand profiles and peaking factors to construct water delivery network infrastructure. The role of smart metering is increasingly recognized by water utilities in demand management, customer service, work optimization and operating efficiency. Today's advanced programs include water-efficient sensors and innovations such as the Internet of Things (IoT) based smart meters are increasingly highlighted. There are significant advances in optimizing water-intensive processes and controlling activities where automatic leak detection and monitoring systems permit to locate and cut off the leaks flow automatically and even patch the leaks.

A. Existing Method

The software combines pulse width ratio and amplitude voltage measurement technology to make the water meter have self-diagnosis function, avoid measurement errors caused by accidental factors, improve adaptability and measurement accuracy, also the application of adaptive measurement period method is used to improve measurement repeatability. The software adopts the moving average filtering algorithm, which effectively reduces the fluctuation of the time-of-flight(ToF) difference and improve the measurement accuracy of the flow point in the low zone (between the minimum flow and the boundary flow).

Problem Statement

- The measurement based on ultrasonic waves.
- Echo pulses varied accordance with the water process as well as pressure of water flow.
- Error data predicted.

B. Proposed Method

Water is a precious need of our lives. Due to the rapid population and urbanization, water usage monitoring is a significant problem facing our society. One solution is to control, analyze, and reduce the water consumption of the houses. The emerging of the Internet of Things (IoT) concept lately in our lives has offered the opportunity to establish water usage-efficient smart devices, systems and applications for buildings and cities. Many studies have suggested designing an IoT-based smart meter system; however, the IoT sensor node has limited studies, especially in battery life.

2. LITERATURE REVIEW

1. (Kiran M. Dhobale , Sangmeshwar P at : 2005)

Water supply management system needs data regarding water storage present in Dam.Satisfying the increasing demand for water supply has been major challenge for many countries around the world. Water is one of the major requirements for human survival, conservation and management of the water resources must be given most importance. The system can measure the water level and give measurement report to the central office. This system use sensors to measure the water level of Dam and updates are provided to Corporation on daily basis.

2. (D.Anandhavalli, K.S. Sangeetha . at : 2003).

Smart water meter is a device that measures the amount of water consumed by householders who have the device fitted within their premises. Water conservation is a big issue in many apartments. A common meter is fitted and cumulative consumption amount is shared among households where they are being charged more than what is to be paid. There are several idea to overcome this issue. In this paper we have proposed a solution to this issue in which a device is used to calculate the flow rate and quantity of water consumed by the householders and send it to the cloud to monitor the consumption of water.

3. (Aritra Ray, Shreemoyee Goswami . at : 2010).

This paper focuses on the developmental and implementation methodology of smart water meter based on Internet of Things (IoT) and Cloud computing equipped with machine learning algorithms, to differentiate between normal and excessive water usage at industrial, domestic and all other sectors having an abundance of water usage, both for Indian and worldwide context. Recognizing that intelligent metering of water has the potential to alter customer engagement of water usage in urban and rural water supplies, this paper fosters for sustainable water management, a need of the present.

3. SYSTEM FUNCTION

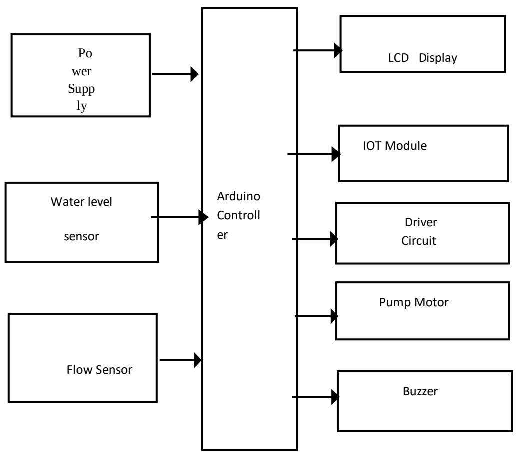

Fig: 1 Block diagram of the System

In this project we proposed the solution for water utilization using water flow sensor and interface with Node MCU microcontroller which embedded with Arduino code. Arduino software is used for Arduino coding to find flow rate of water, display the output in serial monitor and send the sensed data to the cloud which can be monitored by mobile. The prototype model consist of flow sensor, arduino micro controller and IOT module. The flow sensor will interface with controller to monitor the water flow through the Pipe line and monitor the quantity base on the time of water flow. And also to monitor the leakage water.

HARDWARE REQUIREMENTS:

- Power supply
- Flow sensor
- Iot module
- Arduino microcontroller
- Pump motor
- Buzzer

SOFTWARE REQUIREMENTS:

- Arduino IDE
- Embedded C Language

HARDWARE DETAILS:

Transformer:

The potential transformer will step down the power supply voltage (0-230V) to (0-6V) level. Then the secondary of the potential transformer will be connected to the precision rectifier, which is constructed with the help of op–amp. The advantages of using precision rectifier are it will give peak voltage output as DC, rest of the circuits will give only RMS output.

Bridge rectifier:

When four diodes are connected as shown in figure, the circuit is called as bridge rectifier. The input to the circuit is applied to the diagonally opposite corners of the network, and the output is taken from the remaining two corners.

Let us assume that the transformer is working properly and there is a positive potential, at point A and a negative potential at point B. the positive potential at point A will forward bias D3 and reverse bias D4.

The negative potential at point B will forward bias D1 and reverse D2. At this time D3 and D1 are forward biased and will allow current flow to pass through them; D4 and D2 are reverse biased and will block current flow.

IC voltage regulators:

Voltage regulators comprise a class of widely used ICs. Regulator IC units contain the circuitry for reference source, comparator amplifier, control device, and overload protection all in a single IC. IC units provide regulation of either a fixed positive voltage, a fixed negative voltage, or an adjustably set voltage. The regulators can be selected for operation with load currents from hundreds of milli amperes to tens of amperes, corresponding to power ratings from milli watts to tens of watts.

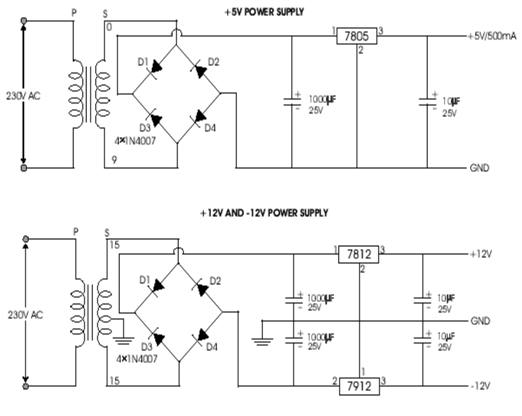

Fig 2 Circuit Diagram Of Power Supply

A fixed three-terminal voltage regulator has an unregulated dc input voltage, Vi, applied to one input terminal, a regulated dc output voltage, Vo, from a second terminal, with the third terminal connected to ground.

The series 78 regulators provide fixed positive regulated voltages from 5 to 24 volts. Similarly, the series 79 regulators provide fixed negative regulated voltages from 5 to 24 volts.

- For ICs, microcontroller, LCD 5 volts.

For alarm circuit, op-amp, relay circuit----------12 volts.

Water Flow Sensor:

Huge industrial plants, commercial and residential buildings require a large amount of water supply. The public water supply system is used to meet this requirement. To monitor the amount of water being supplied and used, the rate of flow of water has to be measured. Water flow sensors are used for this purpose.

Fig 3 Flow Sensor

Working Principle:

Water flow sensor consists of a plastic valve from which water can pass. A water rotor along with a Hall Effect sensor is present the sense and measure the water flow. When water flows through the valve it rotates the rotor. By this, the change can be observed in the speed of the motor. This change is calculated as output as a pulse signal by the hall effect sensor. Thus, the rate of flow of water can be measured.The main working principle behind the working of this sensor is the Hall effect. According to this principle, in this sensor, a voltage difference is induced in the conductor due to the rotation of the rotor. This induced voltage difference is transverse to the electric current.

When the moving fan is rotated due to the flow of water, it rotates the rotor which induces the voltage. This induced voltage is measured by the hall effect sensor and displayed on the LCD display. The water flow sensor can be used with hot waters, cold waters, warm waters, clean water, and dirty water also. These sensors are available in different diameters, with different flow rate ranges. These sensors can be easily interfaced with microcontrollers like Arduino.

Applications Of Water Flow Sensor:

Water flow sensors can measure the rate of flow of water either by measuring velocity or displacement. These sensors can also measure the flow of water like fluids such as measuring milk in a dairy industry etc… There are various types of water flow sensors available based on their diameter and method of measuring.

A cost-effective and most commonly used water flow sensor is Paddlewheel sensor. It can be used with water-like fluids. For the type of applications where a straight pipe is not available for inlet, Positive displacement flow meter is used. This type of water flow sensor can be used for viscous liquids also. For working with dirty water and wastewater which may be conductive, Magnetic flow meter is used. For applications such as sewage water, slurries, and other dirty liquids Ultrasonic flow meters are used.

Arduino UNO Controller:

The Arduino Uno is a microcontroller board based on the ATmega328.It has 14 digital input/output pins (of which six can be used as PWM outputs), six analog inputs, a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with an AC-to-DC adapter or battery to get started. The Arduino Uno differs from all preceding boards because it does not use the FTDI USB-to-serial driver chip. Instead, it features the ATmega8U2 programmed as a USB-to-serial converter.

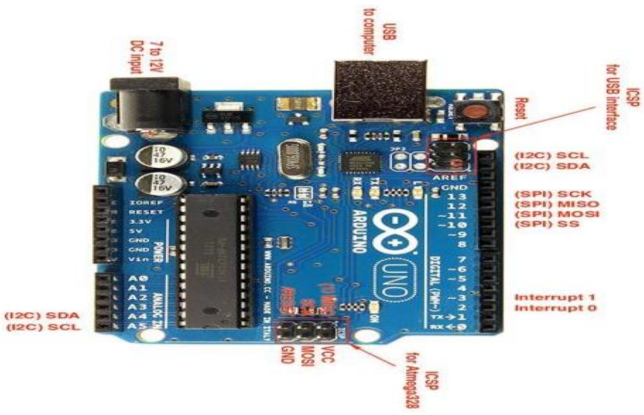

Fig 4 Arduino Uno Controller

The Arduino Uno differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it features the Atmega8U2 microcontroller chip programmed as a USB-to-serial converter. "Uno" means one in Italian and is named to mark the upcoming release of Arduino 1.0. The Arduino Uno and version 1.0 will be the reference versions of Arduno, moving forward. The Uno is the latest in a series of USB Arduino boards, and the reference model for the Arduino platform.

Controlling a Small DC Motor:

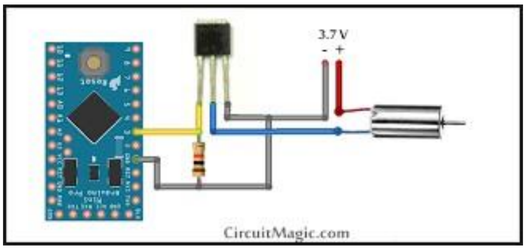

Fig 5 Controlling a Small DC Motor

The connections can be soldered or they can be made through a solderless breadboard. Pin 2 can be any digital I/O pin on your Arduino. Connect the minus of the battery to the emitter of the transistor (E pin) and also connect the emitter of the transistor to Gnd on the Arduino board. To check if things are working, take a jumper wire and short the collector to the emitter pins of the transistor. The motor should turn on. Next, disconnect the 1K resistor from pin 2 and jumper it to +5V. The motor should turn on. Put the resistor back into pin 2 and run the following test program

Buzzer:

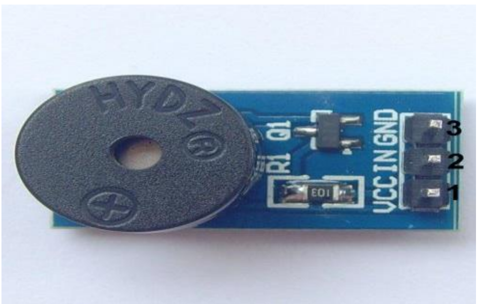

Fig 6 Controlling a Small DC Motor

A buzzer or beeper is a signalling device, usually electronic, typically used in automobiles, household appliances such as a microwave oven, or game shows. It most commonly consists of a number of switches or sensors connected to a control unit that determines if and which button was pushed or a preset time has lapsed, and usually illuminates a light on the appropriate button or control panel, and sounds a warning in the form of a continuous or intermittent buzzing or beeping sound. Initially this device was based on an electromechanical system which was identical to an electric bell without the metal gong (which makes the ringing noise).

SPECIFICATIONS:

- On-board passive buzzer.
- On-board 8550 triode drive.
- Can control with single-chip microcontroller IO directly.
- Working voltage: 5V
- Board size: 22 (mm) x12 (mm)

4. PROGRAMMING CONCEPTS

Embedded 'C':

Embedded C is a set of language extensions for the C Programming language by the C Standards committee to address commonality issues that exist between C extensions for different embedded systems. Historically, embedded C programming requires nonstandard extensions to the C language in order to support exotic features such as fixed-point arithmetic, multiple distinct memory banks, and basic I/O operations.

In 2008, the C Standards Committee extended the C language to address these issues by providing a common standard for all implementations to adhere to. It includes a number of features not available in normal C, such as, fixed- point arithmetic, named address spaces, and basic I/O hardware addressing. Embedded C use most of the syntax and semantics of standard C, e.g., main () function, variable definition, data type declaration, conditional statements (if, switch. case), loops (while, for), functions, arrays and strings, structures and union, bit operations, macros, unions ,etc. As time progressed, use of microprocessor-specific assembly-only as the programming language reduced and embedded systems moved onto C as the embedded programming language of choice. C is the most widely used programming language for embedded processors/controllers. Assembly is also used but mainly to implement those portions of the code where very high timing accuracy, code size efficiency, etc. are prime requirements.

Embedded systems are programmed using different type of languages:

- Machine Code.
- Low level language, i.e., assembly.
- High level language like C, C++, Java, Ada, etc.
- Application level language like Visual Basic, scripts, Access, etc.

Assembly language maps mnemonic words with the binary machine codes that the processor uses to code the instructions. Assembly language seems to be an obvious choice for programming embedded devices. However, use of assembly language is restricted to developing efficient codes in terms of size and speed. Also, assembly codes lead to higher software development costs and code portability is not there. Developing small codes are not much of a problem, but large programs/projects become increasingly difficult to manage in assembly language. Finding good assembly programmers has also become difficult nowadays. Hence high level languages are preferred for embedded systems programming.

Arduino IDE:

All sequential programming languages have four categories of instructions. First are operation commands that evaluate an expression, perform arithmetic, toggle states of I/O lines, and many other operations. Second are jump commands that cause the program to jump immediately to another part of the program that is tagged with a label. Jumps are one way to break out of the normal line-by-line processing mode. For example, if we want a program to repeat over and over without stopping, have the last line of the program be a jump command that takes the program back to its first line. Third are branch commands that evaluate a condition and jump if the condition is true. For example, you might want to jump only if a number is greater than zero. Or, we might want to jump only if the state of an i/o line is low. Fourth are loop commands that repeat a section of code a specified number of times. For example, with a loop we can have a light flash on and off exactly six times.

Arduino Programming Language:

The Arduino runs a simplified version of the C programming language, with some extensions for accessing the hardware. In this guide, we will cover the subset of the programming language that is most useful to the novice Arduino designer. For more information on the Arduino language, see the Language Reference section of the Arduino web site, All Arduino instructions are one line. The board can hold a program hundreds of lines long and has space for about 1,000 two-byte variables. The Arduino executes programs at about 300,000 source code lines per sec.

Program Formatting and Syntax:

Programs are entered line by line. Code is case sensitive which means "myvariable" is different than "MyVariable". Statements are any command. Statements are terminated with a semi-colon. A classic mistake is to forget the semi-colon so if your program does not compile, examine the error text and see if we forgot to enter a colon. Comments are any text that follows "//" on a line. For multi-line block comments, begin with "/*" and end with "*/"

Constants are fixed numbers and can be entered as ordinary decimal numbers (integer only) or in hexadecimal (base 16) or in binary (base 2) as shown in the table below Labels are used to reference locations in your program. They can be any combination of letters, numbers and underscore (_), but the first character must be a letter. When used to mark a location, follow the label with a colon. When referring to an address label in an instruction line, don't use the colon. Here's an example

```
repeat:

 digitalWrite(2,HIGH);

delay(1000);

digitalWrite(2,LOW);

delay(1000);

goto repeat;
```

Symbols are used to redefine how something is named and can be handy for making the code more readable. Symbols are defined with the "#define" command and lines defining symbols should go at the beginning of your program. Here's an example without symbols for the case where an LED is connected to pin 2.

```
void setup()

{

pinMode(2,OUTPUT);

}

void loop()

    }

digitalWrite(2,HIGH); // turn LED on delay(1000);

digitalWrite(2,LOW); // turn LED off delay(1000);

}
```

Here is the same using a symbol to define "LED" #define LED 2 // define the LED pin

```
void setup()

{ pinMode(LED,OUTPUT);

}

void loop()

{

digitalWrite(LED,HIGH); delay(500); digitalWrite(LED,LOW); delay(500);

}
```

Note how the use of symbols reduces the need for comments. Symbols are extremely useful to define for devices connected to pins because if you have to change the pin that the device connects to, you only have to change the single symbol definition rather than going through the whole program looking for references to that pin.

5. HOW TO TEST

1. Connect your Arduino microcontroller to the computer.
2. Connect the VCC pin of your module to the to the 5V pin of your Arduino.
3. Connect the GND pin of your modul to the GND pin of your Arduino.

4. Connect the Input pin of your module to the pin 13 of your Arduino.
5. Enter this program to your Arduino Integrated Development .

```
int buzzer = 13;

void setup()

{

pinMode(buzzer, OUTPUT);

}

void loop()

{

digitalWrite(buzzer, HIGH);

delay(1000);

 digitalWrite(buzzer, LOW);

delay(1000);

}
```

6. TESTING RESULTS

- The sample sketch above is a blink which is also applicable for LEDs.
- The output is the turning on and off of the buzzer every other second.The picture below shows the setup of your module and Arduino:

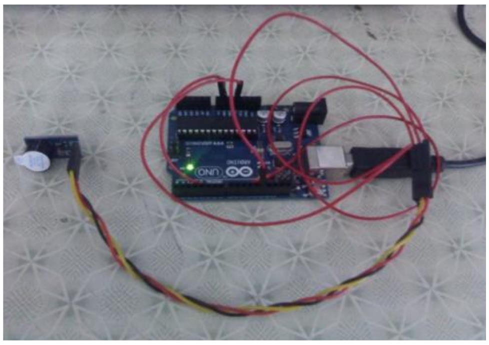

Fig 7. Testing Result

7. CONCLUSION

On the basis of analysis and design, the system provides a smart water meter with eco - friendly and energy efficient system. As the smart water meters are digitized and automated, high accuracy is maintained by decreasing human efforts. Water theft can be avoided since there are no mechanical parts that can be subjected to tamper. A flow sensor based water metering system was used for automated billing, eliminating the drawbacks of traditional water metering systems. Further, multiple houses in a building could use separate end nodes with a common gateway connecting to the internet for accurate billing based on individual consumption of houses. An analysis of water usage through various outlets in a house was provided in order to educate residents on cutting down wasteful usage.

In project phase I , literature review on smart wireless water meter with web DB IOT has been done. Then proposed system block diagram has been studied. Hardware implementation of block diagram will be carried out and result will be discussed in project phase II .

REFERENCES

[1] 1 . Gurung, Ram, Stewart, Rodney, Beal, Cara, K. Sharma, Ashok" Smart meter enabled water end-use demand data: platform for the enhanced infrastructure planning of contemporary urban water supply networks" Published at Journal of Cleaner Production, 2014.

[2] 2. G Hauber-Davidson, E Idris," Smart Water Metering", Published at journal of Water, 2006.

[3] Sarah Darby, "Smart metering: what potential for householder engagement? "Published at BUILDING RESEARCH & INFORMATION,UK, 2010

[4] Tracy C. Britton a, Rodney A. Stewart b, *, Kelvin R. O'Halloran c," Smart metering: enabler for rapid and effective post meter leakage identification and water loss management Tracy" Published at a Journal of Cleaner Production, 2013.

[5] Idris, E. "Smart Metering: A Significant Component of Integrated Water Conservation Systems." In Proceedings of the 1st Australian Young Water Professionals Conference, Sydney, Australia, 2006.

[6] Cara D.BealRodney, A.StewartKellyFielding"A novel mixed method smart metering approach to reconciling differences between perceived and actual residential end use water consumption" Published at Journal of Cleaner Production, 2013,